# Homework — Stage 15: Orchestration & System Design

## Assignment Overview

In the lecture, we learned **how to frame pipelines as DAGs, identify dependencies, and refactor notebook code into modular, CLI-callable functions with logging and checkpoints**. **Now, you will adapt those ideas** to produce (a) a concise orchestration plan for your project and (b) optionally refactor one step of your project into a script-ready function.

## Deliverables

1. **Orchestration Plan (Required)** — 1–2 pages (PDF or Markdown):
    - Jobs/tasks in your project
    - Order/dependencies (DAG sketch allowed as image)
    - Inputs/outputs for each task (file paths or interfaces)
    - Logging & checkpoint strategy
    - What to automate now vs. keep manual (with rationale)
2. **Refactor Demo (Optional Stretch)**:
    - One reusable function with CLI wrapper and simple logging
    - A short README snippet (5–10 lines) explaining how to run it

## Step-by-Step Instructions

1. List 4–8 key tasks in your project (e.g., ingest → clean → model → report).
2. Draw dependencies (arrows). Note parallelizable tasks.
3. For each task, specify:
    - Input(s), Output(s), **idempotency** (Y/N; why)
    - Where to **log** and where to **checkpoint**
4. Identify failure points and preferred **retry** policy (if any).
5. Decide what is worth automating immediately and what is not.
6. (Optional) Refactor one task into a function + CLI wrapper (use the starter).

## Grading Rubric (100 pts total)

- **Pipeline Decomposition (25 pts):** Clear tasks and boundaries (I/O).
- **Dependencies (20 pts):** Correct ordering and rationale.
- **Reliability Plan (20 pts):** Logging, checkpoints, retry thinking.
- **Right-Sizing (15 pts):** Sensible automation decisions for scope.
- **Presentation (10 pts):** Organized, concise, readable diagram.
- **Optional Stretch (+10 pts):** Working function + CLI wrapper.

## Example Submission Expectations

- A simple diagram (photo or draw.io export) of your DAG.
- A table listing tasks with inputs/outputs and logging locations.
- A short paragraph on what to automate now vs. later and why.
- If doing the stretch: a code snippet and how to run it.

## Submission

- Submit to LMS or Git repo.
- Upload: `orchestration_plan.(pdf|md)` and optional code as `refactor_demo/`.

## Constraints & Scope

- Use **only skills from this course**: Python functions, file I/O, logging, CLI basics.
- Do **not** attempt to implement Airflow/Prefect unless already familiar — not required.

---